Lez 7 maggio

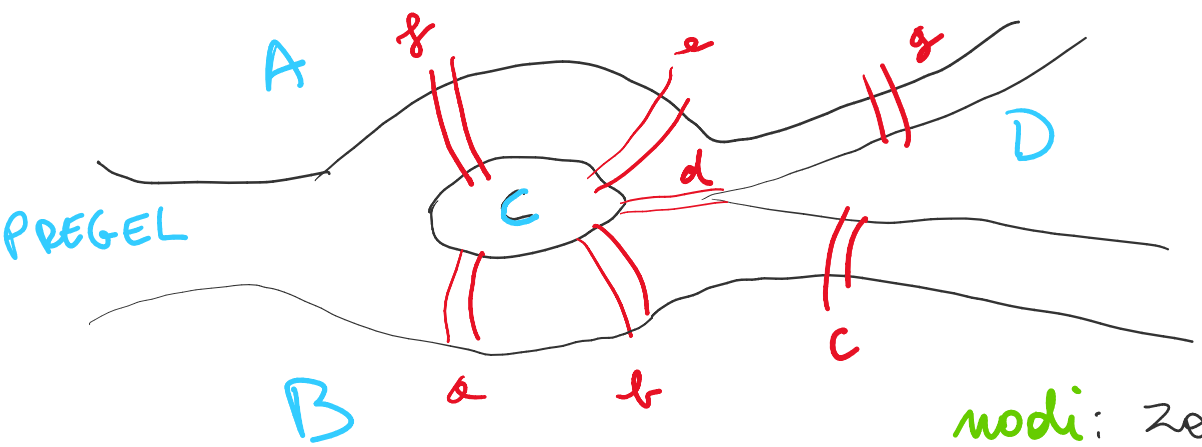
Thursday, May 7, 2020 4:03 PM

Grafo $G = (V, E)$ $(\overrightarrow{v_i, v_j})$ Rete

V insieme dei nodi o vertici $(v_1 \dots v_n)$

E insieme degli archi o spigoli $((v_i, v_j)(v_e, v_m) \dots)$

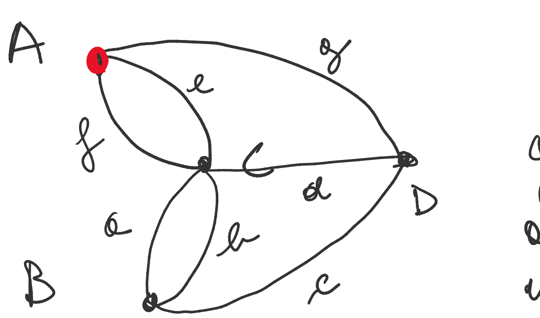
EULERO



nodi: zone
archi: ponti

ALBERTINA PRUSSIA
KÖNIGSBERG

CICLO EULERIANO NO



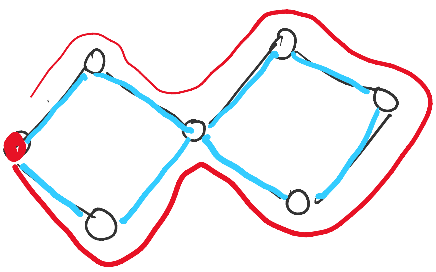
grafo
grado(x) = numero archi incidenti in x

. È possibile partire da un nodo attraversare tutti gli archi del grafo una sola volta tutti gli archi del grafo una sola volta e tornare al nodo di partenza?

- Il ciclo esiste se e solo se tutti i nodi hanno grado pari.
- Algoritmo polinomiale di soluzione

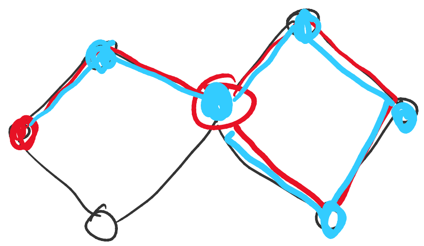
HAMILTON : Ciclo Hamiltoniano

grafo G esiste un ciclo che attraversa tutti i nodi una sola volta.



EULERIANO

FACILE

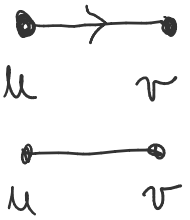


HAMILTONIANO

NO

DIFFICILE

NP-completo



orientati (nodi, archi)

non orientati (vertici, spigoli)

$G = (V, E, W)$ V insieme dei nodi $|V| = n$

E archi $m = |E|$

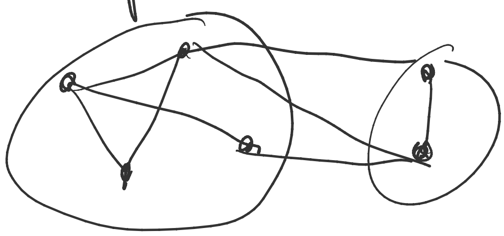
$W : E \to \mathbb{R}$ peso

$\Theta(n)$ grafo sparso

$m = \Theta(n)$ grafo sparso

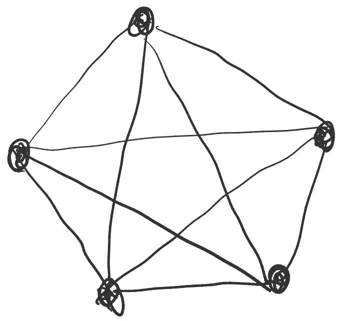
$m = \Theta(n^2)$ grafo denso

grafo <u>connesso</u> se per ogni coppia di nodi esiste un percorso di archi che li collega



$$n-1 \le m \le \frac{n(n-1)}{2} = \Theta(n^2)$$

albero — clique (grafo completo)

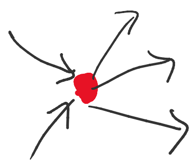


5-CLIQUE

(u, v) u e v sono adiacenti e l'arco è incidente in u e v

grado(x) = numero archi incidenti

IN-DEGREE = 2



OUT-DEGREE = 3

u, v : esiste un <u>cammino</u> da u a v

se esiste $x_0, x_1, \dots, x_k$: $x_0 = u$, $x_k = v$

$(x_i, x_{i+1}) \in E$, $0 \leq \forall i < k$

k lunghezza del cammino

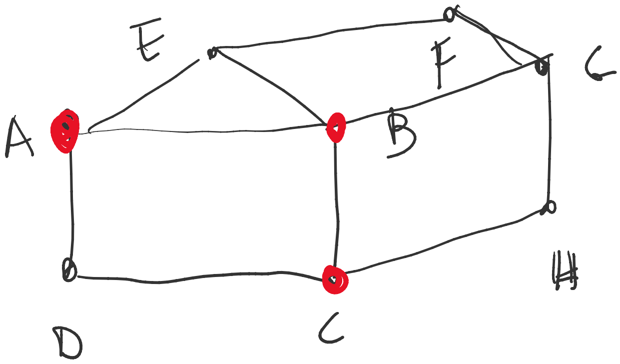
archi pesati: lunghezza = somma dei pesi sugli archi

<u>distanza</u> = cammino minimo tra 2 nodi

<u>sottografo</u> di $G=(V,E)$ è $G'=(V',E')$:
$V' \subseteq V$ e $E' \subseteq E$

Rappresentare i grafi in memoria

- Matrice di adiacenza
- Liste di adiacenza



	A	B	C	D	E	F	G	H
A		1		1	1			

$M(i,j) = 1$

se esiste arco

M

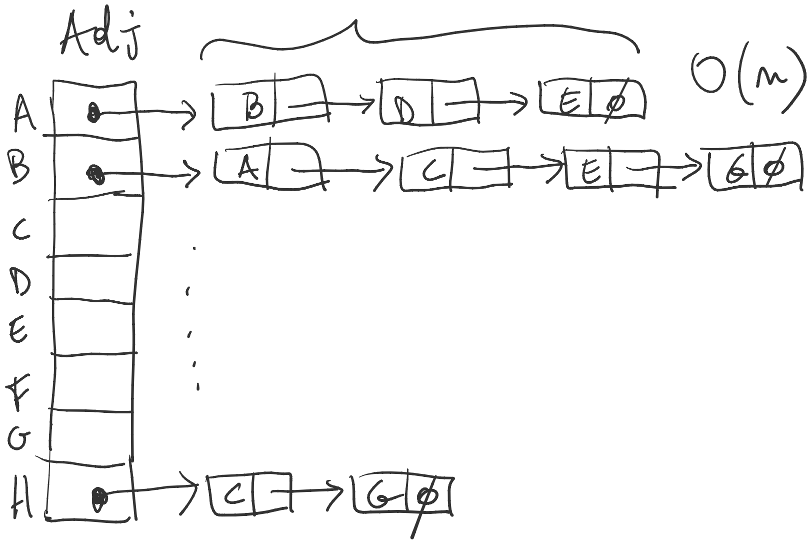
A		1		1	1			
B	1		1		1		1	
C		1		1				1
D								
E								
F								
G								
H								

se esiste arco (i,j)

simmetrica rispetto alla diagonale

Spazio $\underline{\Theta(n^2)}$

Liste di adiacenza $\quad m = \Theta(n^2)$



$O(m)$

$l_1 = 3$

$l_2 = 4$

$l_8 = 2$

(a,b)

(b,a)

Array di puntatori $\rightarrow$ accesso diretto al nodo

adiacenti(H)

$\Theta(1)$ accesso ad H

scansione lista di H di lunghezza l_8

per grafi statici

Spazio $\quad \Theta(n + 2m) = \boxed{\Theta(n+m)}$

archi non orientati

spazio $\Theta(n + 2m) = \boxed{\Theta(\;\;)}$ grafi non orientati

$l_1 + l_2 + \dots + l_n$

$\boxed{\Theta(n + \underline{m})}$ in grafi orientati

$(\overrightarrow{u,v})$ grafo denso $\Theta(n^2)$

grafo sparso $m = \Theta(n)$

$\Downarrow$

Liste di adiacenza spazio $\Theta(n)$

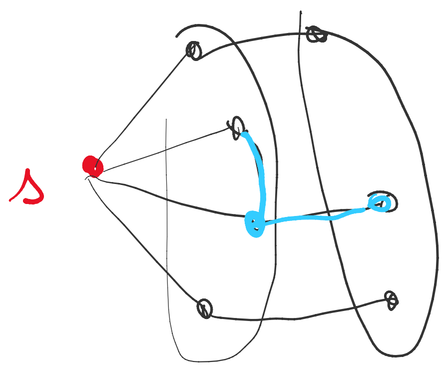
Visita di un grafo

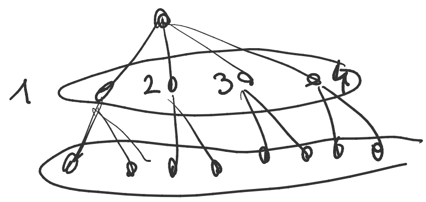
BFS Visita in ampiezza

DFS Visita in profondità

Breadth First Search BFS

Depth First Search DFS





Coda FIFO

bianco inizialmente

u.color < bianco — inizialmente
grigio — quando viene scoperto
nero — quando completa la visita

NULL : u.π = predecessore di u nella visita

0 : u.d = distanza dalla sorgente

BFS (G, s):

<u>for</u> ogni vertice u ∈ V - {s} {
    u.color = white;
    u.d = 0;
    u.π = NULL;
} s.color = gray;

INIZIALIZ.

INIZIO VISITA

s.π = NULL; s.d = 0; Q = ∅;
ENQUEUE (Q, s);
<u>while</u> (Q ≠ ∅) {
    u = DEQUEUE(Q);
    <u>for</u> (ogni vertice v ∈ Adj[u]) {
        if (v.color == white) {
            v.color = gray;
            v.d = u.d + 1;
            v.π = u;

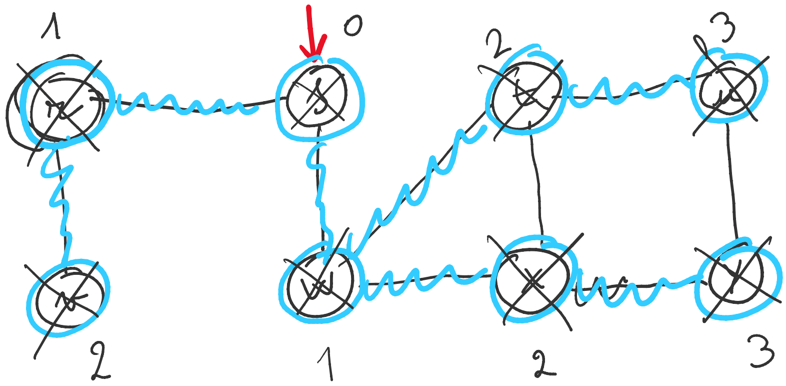
$v.\pi = u;$

$\text{ENQUEUE}(Q, v);$

}

} $u.color = black;$

}



s: r, w

r: v

t: u, x, w

u: t, y

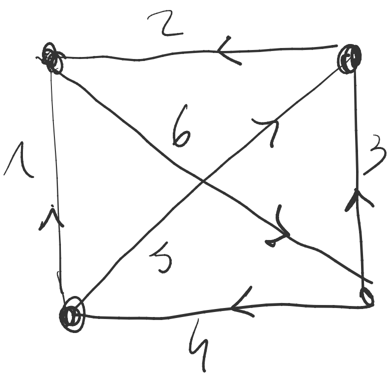
w: s, t, x

x: t, w, y

y: u, x

$Q = \{\cancel{s}, \cancel{r}, \cancel{w}, \cancel{v}, \cancel{t}, \cancel{x}, \cancel{u}, \cancel{y}\}$

s r w v t x u y



4-clique

$$\frac{n(n-1)}{2} = \frac{4 \cdot 3}{2} = 6 \qquad \Theta(n^2)$$